| CS 598CSC: Approximation Algorithms | Lecture date: January 26, 2011 |
|---|---|
| Instructor: Chandra Chekuri | Scribe: Md. Abul Hassan Samee (2009) |

Notes edited by instructor in 2011.

# 1 Introduction

We discuss two closely related NP Optimization problems, namely Set Cover and Maximum Coverage in this lecture. Set Cover was among the first problems for which approximation algorithms were analyzed. This problem is also significant from a practical point of view, since the problem itself and several of its generalizations arise quite frequently in a number of application areas. We will consider three such generalizations of Set Cover in this lecture. We conclude the lecture with a brief discussion on how the Set Cover problem can be formulated in terms of submodular functions.

# 2 Set Cover and Maximum Coverage

## 2.1 Problem definition

In both the Set Cover and the Maximum Coverage problems, our input is a set $\mathcal{U}$ of $n$ elements, and a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of $m$ subsets of $\mathcal{U}$ such that $\bigcup_i S_i = \mathcal{U}$. Our goal in the Set Cover problem is to select as few subsets as possible from $\mathcal{S}$ such that their union covers $\mathcal{U}$. In the Maximum Coverage problem an integer $k \leq m$ is also specified in the input, and our goal is to select $k$ subsets from $\mathcal{S}$ such that their union has the maximum cardinality. Note that the former is a minimization problem while the latter is a maximization problem. One can also consider weighted versions of these problems which we postpone to a later lecture.

## 2.2 Greedy approximation

Both Set Cover and Maximum Coverage are known to be **NP**-Hard. A natural greedy approximation algorithm for these problems is as follows.

```
Greedy Cover (U, S):
1: repeat
2:         pick the set that covers the maximum number of uncovered elements
3:         mark elements in the chosen set as covered
4: until done
```

In case of Set Cover, the algorithm Greedy Cover is *done* in line 4 when all the elements in set $\mathcal{U}$ have been covered. And in case of Maximum Coverage, the algorithm is *done* when exactly $k$ subsets have been selected from $\mathcal{S}$.

## 2.3 Analysis of Greedy Cover

**Theorem 1** Greedy Cover *is a* $1 - (1 - 1/k)^k \geq (1 - \frac{1}{e}) \simeq 0.632$ *approximation for* Maximum Coverage*, and a* $(\ln n + 1)$ *approximation for* Set Cover.

The following theorem due to Feige [1] implies that GREEDY COVER is essentially the best possible in terms of the approximation ratio that it guarantees in Theorem 1.

**Theorem 2** *Unless* $\boldsymbol{NP} \subseteq \boldsymbol{DTIME}(n^{O(\log\log n)})$*, there is no* $(1-o(1))\ln n$ *approximation for* SET COVER*. Unless* $\boldsymbol{P}=\boldsymbol{NP}$*, for any fixed* $\epsilon > 0$*, there is no* $(1-\frac{1}{e}-\epsilon)$ *approximation for* MAXIMUM COVERAGE*.*

We proceed towards the proof of Theorem 1 by providing analysis of GREEDY COVER separately for SET COVER and MAXIMUM COVERAGE. Let $OPT$ denote the value of an optimal solution to the MAXIMUM COVERAGE problem. Let $x_i$ denote the number of *new* elements covered by GREEDY COVER in the $i$-th set that it picks. Also, let $y_i = \sum_{j=1}^{i} x_i$, and $z_i = OPT - y_i$. Note that, according to our notations, $y_0 = 0$, $y_k$ is the number of elements chosen by GREEDY COVER, and $z_0 = OPT$.

## Analysis for MAXIMUM COVERAGE

We have the following lemma for algorithm GREEDY COVER when applied on MAXIMUM COVERAGE.

**Lemma 3** GREEDY COVER *is a* $1-(1-1/k)^k \geq 1-\frac{1}{e}$ *approximation for* MAXIMUM COVERAGE*.*

We first prove the following two claims.

**Claim 4** $x_{i+1} \geq \frac{z_i}{k}$.

**Proof:** At each step, GREEDY COVER selects the subset $S_j$ whose inclusion covers the maximum number of uncovered elements. Since the optimal solution uses $k$ sets to cover OPT elements, some set must cover at least $1/k$ fraction of the at least $z_i$ remaining uncovered elements from OPT. Hence, $x_{i+1} \geq \frac{z_i}{k}$. □

**Claim 5** $z_{i+1} \leq (1-\frac{1}{k})^{i+1} \cdot OPT$

**Proof:** The claim is true for $i = 0$. We assume inductively that $z_i \leq (1-\frac{1}{k})^i \cdot OPT$. Then

$$\begin{aligned} z_{i+1} &\leq z_i - x_{i+1} \\ &\leq z_i(1-\frac{1}{k}) \quad \text{[using Claim 4]} \\ &\leq (1-\frac{1}{k})^{i+1} \cdot OPT. \end{aligned}$$

□

**Proof of Lemma 3.** It follows from Claim 5 that $z_k \leq (1-\frac{1}{k})^k \cdot OPT \leq \frac{OPT}{e}$. Hence, $y_k = OPT - z_k \geq (1-\frac{1}{e}) \cdot OPT$. □

## Analysis for SET COVER

We have the following lemma.

**Lemma 6** GREEDY COVER *is a* $(\ln n + 1)$ *approximation for* SET COVER*.*

Let $k^*$ denote the value of an optimal solution to the SET COVER problem. Then an optimal solution to the MAXIMUM COVERAGE problem for $k = k^*$ would cover all the $n$ elements in set $\mathcal{U}$, and $z_{k^*} \leq \frac{n}{e}$. Therefore, $\frac{n}{e}$ elements would remain uncovered after the first $k^*$ steps of GREEDY COVER. Similarly, after $2 \cdot k^*$ steps of GREEDY COVER, $\frac{n}{e^2}$ elements would remain uncovered. This easy intuition convinces us that GREEDY COVER is a $(\ln n + 1)$ approximation for the SET COVER problem. A more succinct proof is given below.

**Proof of Lemma 6.** Since $z_i \leq (1 - \frac{1}{k^*})^i \cdot n$, after $t = k^* \ln \frac{n}{k^*}$ steps, $z_t \leq k^*$. Thus, after $t$ steps, $k^*$ elements are left to be covered. Since GREEDY COVER picks at least one element in each step, it covers all the elements after picking at most $k^* \ln \frac{n}{k^*} + k^* \leq k^*(\ln n + 1)$ sets. □

The following corollary readily follows from Lemma 6.

**Corollary 7** *If $|S_i| \leq d$, then* GREEDY COVER *is a $(\ln d + 1)$ approximation for* SET COVER.

**Proof:** Since $k^* \geq \frac{n}{d}$, $\ln \frac{n}{k^*} \leq \ln d$. Then the claim follows from Lemma 6. □

**Proof of Theorem 1.** The claims follow directly from Lemma 3 and 6. □

## A tight example for GREEDY COVER when applied on SET COVER

Let us consider a set $\mathcal{U}$ of $n$ elements along with a collection $\mathcal{S}$ of $k+2$ subsets $\{R_1, R_2, C_1, C_2, \ldots, C_k\}$ of $\mathcal{U}$. Let us also assume that $|C_i| = 2^i$ and $|R_1 \cap C_i| = |R_2 \cap C_i| = 2^{i-1}$ $(1 \leq i \leq k)$, as illustrated in Fig. 1.

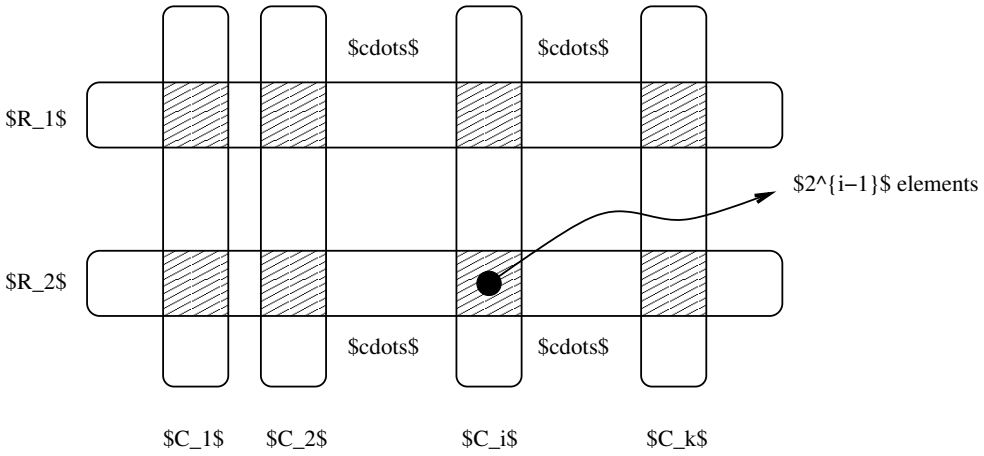


Figure 1: A tight example for GREEDY COVER when applied on SET COVER

Clearly, the optimal solution consists of only two sets, i.e., $R_1$ and $R_2$. Hence, $OPT = 2$. However, GREEDY COVER will pick each of the remaining $k$ sets, namely $C_k, C_{k-1}, \ldots, C_1$. Since $n = 2 \cdot \sum_{i=0}^{k-1} 2^i = 2 \cdot (2^k - 1)$, we get $k \approx \Omega(\log_2 n)$. Hence the example is tight.

**Exercise:** Consider the weighted version of the SET COVER problem where a weight function $w : \mathcal{S} \to \mathcal{R}^+$ is given, and we want to select a collection $\mathcal{S}'$ of subsets from $\mathcal{S}$ such that $\cup_{X \in \mathcal{S}'} X = \mathcal{U}$, and $\sum_{X \in \mathcal{S}'} w(X)$ is the minimum. Prove that the greedy heuristic gives a $2 \cdot (\ln n + 1)$ approximation for this problem.

*Hint 1:* Note that the greedy algorithm never picks a set of cost more than OPT. *Hint 2:* By the first time the total cost of sets picked by the greedy algorithm exceeds OPT, it has covered a $(1 - 1/e)$ fraction of the elements.

# 3 Dominating Set and Vertex Cover

## 3.1 Dominating Set

A *dominating set* in a graph $G = (V, E)$ is a set $S \subseteq V$ such that for each $u \in V$, either $u \in S$, or some neighbor $v$ of $u$ is in $S$. In the Dominating Set problem, our goal is to find a smallest dominating set of $G$.

A natural greedy algorithm for this problem is to iteratively choose a vertex with the highest degree. It can be proved that this heuristic gives a $(\ln n + 1)$, or more accurately, a $(\ln(\Delta + 1) + 1)$ approximation for the Dominating Set problem.

**Exercises:**

1. Prove the approximation guarantees of the greedy heuristic for Dominating Set.
2. Show that Dominating Set is a special case of Set Cover.
3. Show that Set Cover can be reduced in an approximation preserving fashion to Dominating Set. More formally, show that if Dominating Set has an $\alpha(n)$-approximation where $n$ is the number of vertices in the given instance then Set Cover has an $(1 - o(1))\alpha(n)$-approximation.

## 3.2 Vertex Cover

A *vertex cover* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that for each edge $e \in E$, at least one end point of $e$ is in $S$. In the Vertex Cover problem, our goal is to find a smallest vertex cover of $G$. In the *weighted* version of the problem, a weight function $w : V \to \mathcal{R}^+$ is given, and our goal is to find a minimum weight vertex cover of $G$. The unweighted version of the problem is also known as Cardinality Vertex Cover.

It can be shown that, the Greedy Cover algorithm can give an $O(\ln \Delta + 1)$ approximation for both weighted and unweighted versions of the Vertex Cover problem.

**Exercises:**

1. Show that Vertex Cover is a special case of Set Cover.
2. Construct an example that shows that Greedy Cover when applied on the Vertex Cover problem gives an $\Omega(\log n)$-approximation.

### 3.2.1 Better (constant) approximation for Vertex Cover

Cardinality Vertex Cover : The following is a 2-approximation algorithm for the Cardinality Vertex Cover problem.

```
Matching-VC (G):
1: S ← ∅
2: Compute a maximal matching M in G
3: for each edge (u, v) ∈ M do
4:         add both u and v to S
5: Output S
```

**Theorem 8** Matching-VC *is a* 2*-approximation algorithm.*

The proof of Theorem 8 follows from two simple claims.

**Claim 9** *Let $OPT$ be the size of the vertex cover in an optimal solution. Then $OPT \geq |M|$.*

**Proof:** Since the optimal vertex cover must contain at least one end vertex of every edge in $M$, $OPT \geq |M|$. □

**Claim 10** *Let $S(M) = \{u, v | (u, v) \in M\}$. Then $S(M)$ is a vertex cover.*

**Proof:** If $S(M)$ is not a vertex cover, then there must be an edge $e \in E$ such that neither of its endpoints are in $M$. But then $e$ can be included in $M$, which contradicts the maximality of $M$. □

**Proof of Theorem 8.** Since $S(M)$ is a vertex cover, Claim 9 implies that $|S(M)| = 2 \cdot |M| \leq 2 \cdot OPT$. □

Weighted Vertex Cover: 2-approximation algorithms for the Weighted Vertex Cover problem can be designed based on LP rounding or Primal-Dual technique. These will be covered later in the course.

### 3.2.2 Set Cover with small frequencies

Vertex Cover is an instance of Set Cover where each element in $\mathcal{U}$ is in at most two sets (in fact, each element was in exactly two sets). This special case of the Set Cover problem has given us a 2-approximation algorithm. What would be the case if every element was contained in at most three sets? More generally, given an instance of Set Cover, for each $e \in \mathcal{U}$, let $f(e)$ denote the number of sets containing $e$. Let $f = \max_e f(e)$, which we call the *maximum frequency.*

**Exercise:** Give an $f$-approximation for Set Cover, where $f$ isthe maximum frequency of an element. *Hint:* Follow the approach used for Vertex Cover .

# 4 Two important aspects of greedy approximation for Set Cover

## 4.1 Greedy approximation for implicit instances

It turns out that the universe $\mathcal{U}$ of elements and the collection $\mathcal{S}$ of subsets of $\mathcal{U}$ are not restricted to be finite or explicitly enumerated in the Set Cover problem. For instance, a problem could

require covering a finite set of points in the plane using disks of unit radius. There is an infinite set of such disks, but the greedy approximation algorithm can still be applied. For such implicit instances, the greedy algorithm can be used if we have access to an *oracle*, which, at each iteration, selects a set having the optimal density. However, an oracle may not always be capable of selecting an optimal set. In such cases, it may have to make the selections *approximately*. We call an oracle an $\alpha$-*approximate oracle* if, at each iteration, it selects a set $S$ such that $density(S) \geq \alpha \cdot Optimal\ Density$, for some $\alpha > 1$.

**Exercise:** Prove that the approximation guarantee of greedy approximation with an $\alpha$-approximate oracle would be $\alpha(\ln n + 1)$ for Set Cover, and $(1 - \frac{1}{e^{\alpha}})$ for Maximum Coverage.

## 4.2 Greedy approximation for submodular functions

In a more general sense, the greedy approximation works for any *submodular set function*. Given a finite set $E$, a function $f : 2^E \to \mathcal{R}^+$ is submodular iff $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq E$. Alternatively, $f$ is a submodular functions iff $f(A + i) - f(A) \geq f(B + i) - f(B)$ for all $i \in E$ and $A \subset B$. This second characterization is due to the property of *decreasing marginal utility* of submodular functions. Intuitively, adding element $i$ to a set $A$ will help at least as much as adding it to to a (larger) set $B \supset A$.

**Exercise:** Prove that the two characterizations of submodular functions are equivalent.

A submodular function $f(\cdot)$ is *monotone* if $f(A+i) \geq f(A)$ for all $i \in E$ and $A \subseteq E$. We assume that $f(\emptyset) = 0$. Submodular set functions arise in a large number of practical fields including combinatorial optimization, probability, and geometry. Examples include rank function of a matroid, the sizes of cutsets in a directed or undirected graph, the probability that a subset of events do not occur simultaneously, entropy of random variables, etc. In the following we show that the Set Cover and Maximum Coverage problems can be easily formulated in terms of submodular set functions.

**Exercise.** Suppose we are given a universe $\mathcal{U}$ and a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of subsets of $\mathcal{U}$. Now if we take $N = \{1, 2, \ldots, m\}$, $f : 2^N \to \mathcal{R}^+$, and define $f(A) = |\cup_{i \in A} S_i|$ for $A \subseteq E$, then show that the function $f$ is submodular.

### 4.2.1 Submodular Set Cover

When formulated in terms of submodular set functions, the Set Cover problem is the following. Given a monotone submodular function $f$ (whose value would be computed by an oracle) on $N = \{1, 2, \ldots, m\}$, find the smallest set $S \subseteq N$ such that $f(S) = f(N)$. Our previous greedy approximation can be applied to this formulation as follows.

```
Greedy Submodular (f, N):
1: S ← ∅
2: while f(S) ≠ f(N)
3:         find i to maximize f(S + i) − f(S)
4:         S ← S ∪ {i}
```

**Exercises:**

1. Prove that the greedy algorithm is a $1 + \ln(f(N))$ approximation for Submodular Set Cover.

2. Prove that the greedy algorithm is a $1 + \ln(\max_i f(i))$ approximation for Submodular Set Cover.

### 4.2.2 Submodular Maximum Coverage

By formulating the Maximum Coverage problem in terms of submodular functions, we seek to maximize $f(S)$ such that $|S| \leq k$. We can apply algorithm Greedy Submodular for this problem by changing the condition in line 2 to be: **while** $|S| \leq k$.

**Note.** For the Submodular Maximum Coverage problem, function $f$ must be both submodular and monotone.
**Exercise:** Prove that greedy gives a $(1 - 1/e)$-approximation for Submodular Maximum Coverage problem.

## References

[1] U. Feige. A Threshold of $\ln n$ for Approximating Set Cover. *J. of the ACM* 45(5): 634–652, 1998.